

Project Report

MyStore - A Full-Stack E-Commerce Marketplace

Submitted By:

Yogesh Kohli

UID: 23BCS10291

Section: KRG - 3B

Date: November 10, 2025

1. Introduction & Abstract

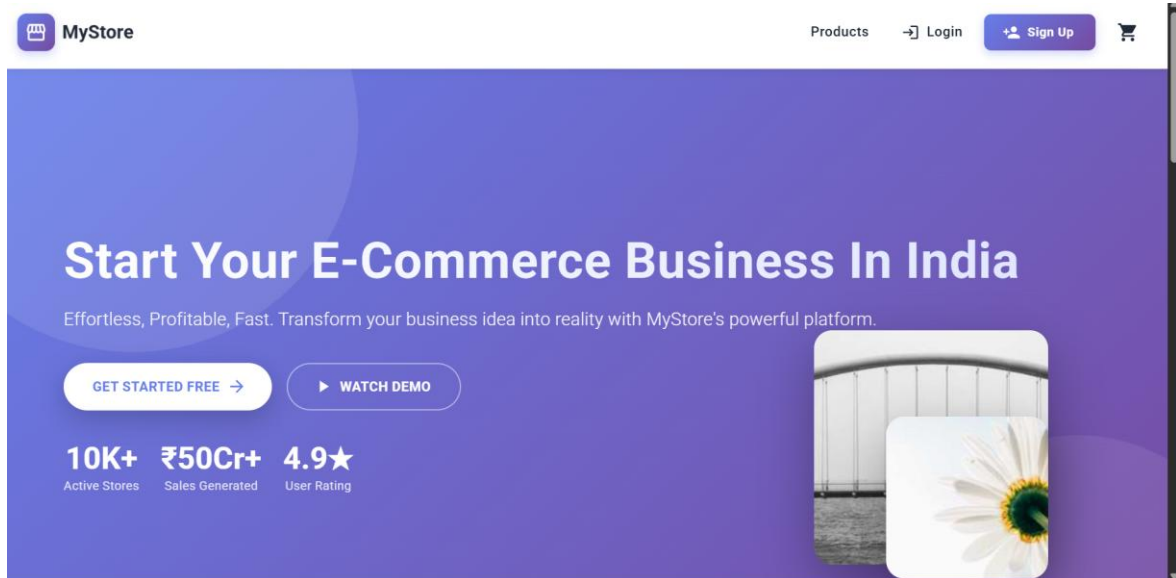
MyStore is a full-stack e-commerce marketplace designed to connect sellers and customers through a seamless digital experience. The platform integrates a responsive React frontend with a secure Spring Boot backend to manage data and operations efficiently.

It supports two main roles: Customers, who can browse and purchase products, and Sellers, who can manage their virtual stores by adding or removing products. The system demonstrates modern software engineering concepts like RESTful API design, database normalization, and responsive UI development.

2. Project Objectives

The objectives of MyStore were to design and develop a complete marketplace application with the following goals:

- Build a secure and scalable backend using Spring Boot.
- Create a responsive and intuitive frontend using React and Material-UI.
- Implement role-based access control for Customers and Sellers.
- Allow sellers to manage (add/delete) their products securely.
- Enable customers to search and filter products efficiently.
- Maintain data consistency using a relational database model with MS SQL Server.



3. System Architecture

MyStore follows a three-tier architecture:

1. Frontend (Client Layer): Built using React, the frontend is responsible for user interaction, UI rendering, and communicating with the backend API via Axios. Material-UI ensures a clean and modern interface adaptable to all screen sizes.
2. Backend (Application Layer): Developed with Spring Boot, the backend handles business logic, processes API requests, and validates data. It provides RESTful services for user authentication, product management, and data retrieval.
3. Database (Data Layer): Microsoft SQL Server stores user, role, and product data. The schema maintains referential integrity and ensures efficient querying through well-structured relationships.

4. Technologies Used

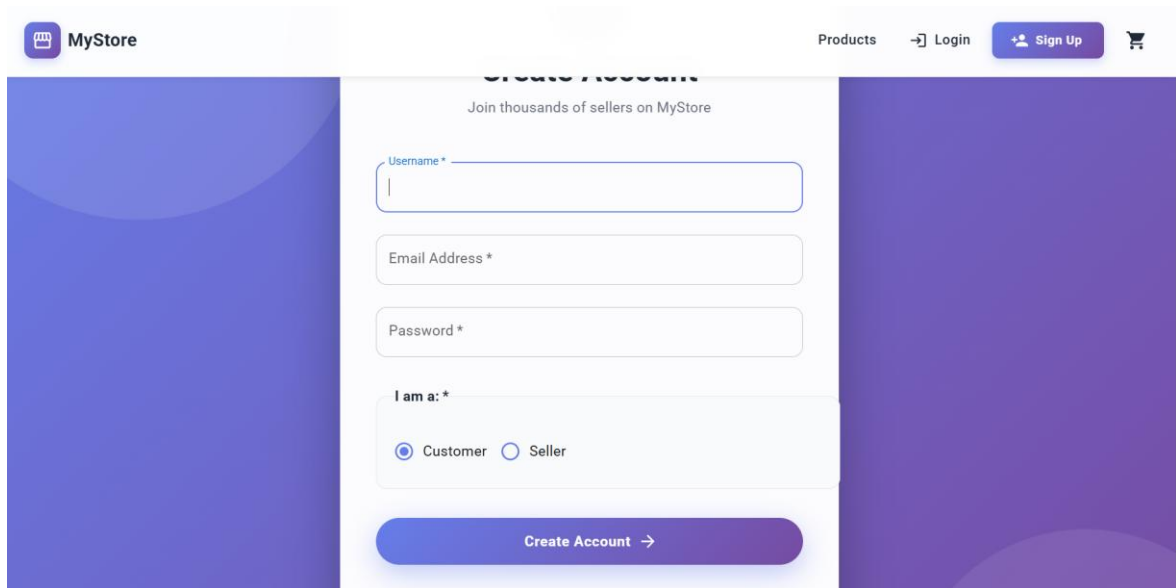
The development of MyStore required a range of modern technologies, each chosen for performance, reliability, and integration ease.

Frontend:

- React: For building dynamic and component-based UIs.
- Vite: For fast builds and hot module reloading.
- Material-UI (MUI): For polished design and responsive layouts.
- Axios: For seamless API communication.
- React Router: For managing navigation without page reloads.

Backend:

- Spring Boot (Java 17): Core framework for the REST API.



- Spring Web & Spring Data JPA: For handling web requests and database interaction.
- Maven: For project management and dependency control.

Database:

- Microsoft SQL Server: For storing structured relational data.
- JDBC Driver: For connecting Java applications to the SQL Server database.

5. Key Features Implemented

1. User Authentication and Role Management:

Users can sign up either as Customers or Sellers. Sellers must provide a unique shop name during registration. Authentication ensures only authorized users access restricted functionalities.

2. Product Management (Seller-Specific):

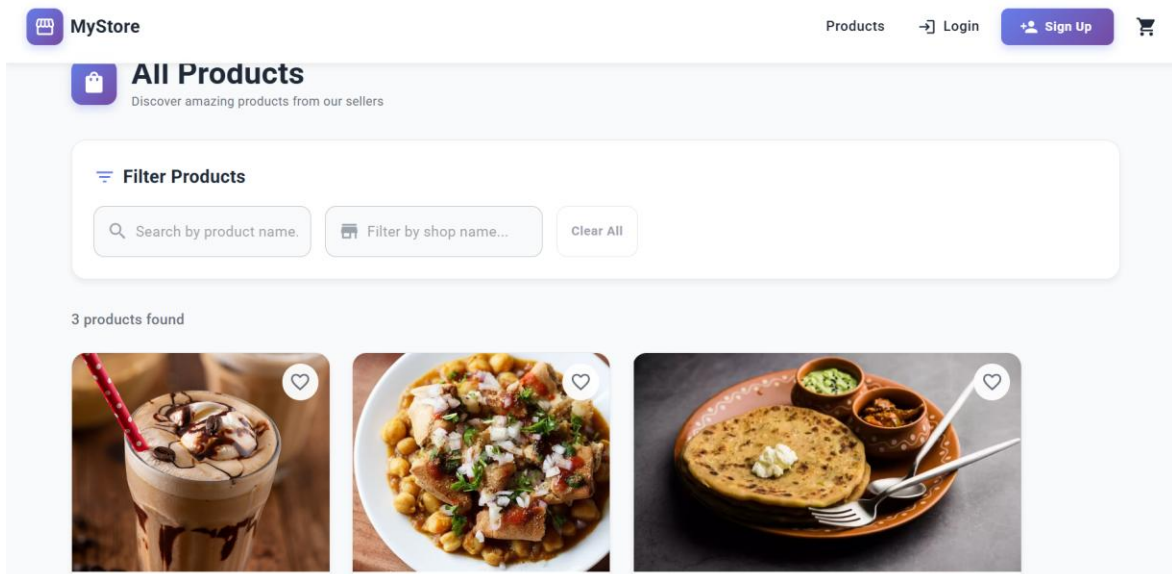
Sellers can add products with details like name, price, description, and image URL. They can also delete only their own products, ensuring secure ownership verification.

3. Product Search and Filtering (Customer-Specific):

Customers can search for products by name or filter them based on the shop name. The system dynamically updates the product list for an enhanced browsing experience.

4. Secure API Access:

Every endpoint is validated with role-based permissions, ensuring data security and integrity across user roles.



6. Database Schema

The database design emphasizes simplicity and relational integrity.

Users Table:

- id: Primary Key
- username: Display Name
- email: Unique email ID
- password: Encrypted password
- role: CUSTOMER / SELLER- shop_name: Nullable, only for sellers

Products Table:

- id: Primary Key
- name: Product name
- description: Product details
- price: Product price
- image_url: Link or Base64 string
- seller_id: Foreign Key referencing Users(id)

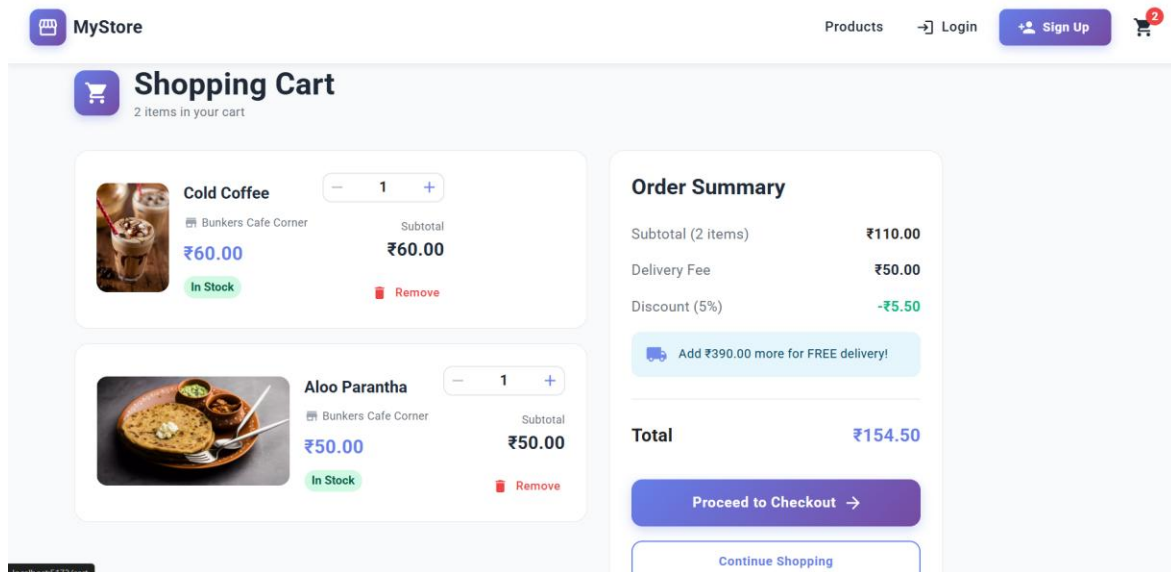
This establishes a One-to-Many relationship — one seller can manage multiple products.

7. Implementation Overview

The frontend communicates with the backend using RESTful APIs. Product-related operations are executed using HTTP methods — GET for fetching, POST for adding, and

DELETE for removing products. The backend performs validation and authorization before executing any operation.

The database interactions are managed by Spring Data JPA, minimizing boilerplate code and improving readability. The application also incorporates form validation, error handling, and asynchronous API calls for smooth functionality.



8. Conclusion and Future Scope

The MyStore project effectively demonstrates end-to-end full-stack web development. It integrates secure backend services with a user-friendly frontend and an optimized database model. The project successfully fulfills its objectives, delivering a scalable and maintainable e-commerce solution.

Future Enhancements:

- Add Shopping Cart and Checkout with payment gateway integration.
- Introduce Product Details Page for better user experience.
- Implement Customer Profiles with purchase history.
- Enable Product Reviews and Ratings.
- Deploy the platform on cloud hosting (e.g., AWS, Render, or Azure).

With these improvements, MyStore can evolve into a production-ready marketplace platform supporting real-world commercial use cases.